Designing *Useful* Privacy Applications

Len Sassaman
Nomen Abditum Services
rabbia@abditum.com
What are Privacy Applications?
Internet Privacy Applications

- Introduce measurable privacy into a system
- Conceal information about the user
- Selectively reveal credentials or attributes
- Restrict access to private information
- Possibly conceal user’s identity
- Prevent unauthorized observation of communication, financial activity, or other sensitive behavior
Examples of PETs

- Encryption applications and protocols
  - PGP
  - S/MIME
  - Disk encryption
  - IPsec
  - SSL/TLS
Examples of PETs

• Local computer security measures
  – Cookie managers
  – P3P clients
  – Personal firewalls
  – Digital wallets
Examples of PETs

• Anonymity services
  – Anonymous remailers
    • Mixmaster
    • Mixminion
  – Anonymous web proxies
    • Anonymizer
    • JAP
  – IP level anonymity
    • TOR
    • Freedom
Are users concerned about privacy?
User demand for privacy

… or lack thereof
• Risks users face
• Credit card fraud (not a risk)
• ID theft?
• Trusting in the law
  – (of math or men?)

• Consumers understand threat analysis!
Ten years ago…

- A bright future for the crypto-utopia
- E-Cash, remailers, and revolution
- Cryptoanarchy and the Cypherpunks
- Empowerment of individuals’ liberties
- Creation of programs – arms for the masses.
Today

What crypto successes have happened?

- SSL/TLS
- PGP (who uses it?)
- S/MIME, PEM, MOSS…
- Disk encryption
- E-cash (hah!)
- Anonymizer vs. Mixmaster
Dismal usage statistics

Usability is a security consideration
10 years of Cryptomasturbation

Was the problem...

- User apathy?
- Developer incompetence?
- Smart people unaware of their audiences?
- Programmers wanting “cool projects”?
- Development decisions based on politics?
- Is usability an intractable problem?
Primary problems

• Lack of perceived need
• Single fax machine problem
• Lack of forced adoption
• Lack of availability
• Lack of competency in forced adoption
• Interoperability
  – backwards compatibility – not always smart
Primary problems

• Poor user interface!!
• Standards bloat
• Developer mentality
  – blinded by details
  – if we can’t have it all, it isn’t worth doing
  – well, a strong system that isn’t used is worthless!
• Did I mention UI?
Where has crypto/privacy improved?

- Where expectations for UI were low
- Where crypto was already used
- Military -- no choice
The Cryptographer and the Locksmith

- Who really understands threat analysis?
- Customers understand, but aren’t qualified to evaluate
- Locksmiths recognize and accept security trade-offs
- Academic cryptographers strive for perfect security
- Implementers need to be more like Locksmiths
Protocol Pitfalls

- Over-extension of protocols
- Addition of functionality to the protocol, rather than addition of protocols to the application
- Complex protocols are hard to audit
- Complex protocols are hard to implement
- Complex protocols are hard to make interoperable
Protocol Pitfalls

• Algorithm choices should be fixed
  – Protocols with parameterized algorithms are more likely to break
  – Incompatibility between implementations
  – Increased chance of compromise
  – Legacy “weak cipher” support
  – Backwards compatibility with implementations that support broken algorithms -- bad!
An Alternative to Parameterization

• Build protocols with single algorithm choices
• Design protocol so that easy replacement of a defective algorithm is possible
• Intentionally break backwards compatibility with weaker protocol versions!
PGP

Our biggest failure
Choice Quotes

• “I get one piece of PGP-encrypted mail every month or two” – Peter Gutmann
• “Going on 9 full years after I generated my first PGP key, my mom still can’t use the stuff.” – Adam Shostack
• “I, too, rarely encrypt.” – Tim May
• “PGP has […] an architectural attitude problem.” – Rodney Thayer
In addition to the other problems:

• What is PGP’s purpose?
• Product name dilution
  – what is PGP?
• Confusing terminology
  – public key, private key, symmetric key, algorithm, cipher, hash, armor, signature, validity, authenticity, fingerprint, footprint, good, bad…
Authentication vs. Encryption

- SSL/TLS adoption is greatly weakened by the belief that these must go together
- PGP is also scary, because of auth issues
- How do we know Bob isn’t Eve?
RFC 2440

- OpenPGP protocol specification
- Encompasses RFC 1991 as well
  - Why? Applications can just implement both
- Multiple symmetric cipher algorithms
  - Let’s talk about what happens if 3DES is broken…
- Multiple hash algorithms
  - Protocol is as secure as the weakest hash
- Protocol extensions leak implementation details
- User preferences are not always honored
The Web of Trust

• “I don’t think that word means what you think it means.”
• A major misnomer
• Really a web of assertions
• What does a signature certify?
  – ID
  – Possibly trust in signee’s CSP
• Major depth limitations
  – “trusted introducers”
How to make a true user-empowering system

- Friendly UI!
- Simplified concepts
- One-click usage
- Better integration
- No room for error
- Proper usage the only usage
- “open-hood architecture”
Existing attempts

• PGP, with the UI/interop exceptions, is a decent system
• Can be used as an underlying protocol
• Hushmail
• Zendit
• Lokmail
Alternatives to PGP

- New OpenPGP Protocol version?
- Application specific crypto
  - Trillian “SecureIM”
- Off the Record Messaging
  - http://www.cypherpunks.ca/otr/
- STARTTLS
New OpenPGP Protocol

- Eliminate legacy issues
- Correct existing flaws
- Select single algorithms
- Backwards compatibility at application level
- Would be similar to existing OpenPGP
  - Intuitive for developers
  - Consistent with existing implementations
  - Same/similar library APIs
Application specific privacy

- Nothing more than is needed
- Simple protocols for specific purposes
- Presumes a closed system for maximal adoption
- Interoperability within system is good
- Interoperability outside system is nonexistent
- Trillian AIM users cannot use crypto with non-Trillian AIM users
- Sametime users cannot use crypto with AIM users
Off the Record Messaging

- Work by Borisov, Goldberg, Brewer
- Adds perfect forward secrecy and repudiability
- Simple protocol with reasonable algorithm choices
- Will work over existing IM systems or email
- Could be given a simple UI
STARTTLS

• Server level encryption
• Doesn’t offer much security against an evil ISP or an active attacker
• Invisible to the user (great UI!)
• Opportunistic
• Low cost
• Can the same model work in the users’ hands?
Acid test

• A good crypto program will have a UI that:
• Needs no manual
• Can consist entirely of icons – no words!
• Requires no more skill than a basic email program
• Does not inconvenience the user
• Adds at most one extra click
Comments

Len Sassaman
rabbi@abditum.com